

# Um Método para Avaliação Automática da Qualidade de Recursos Educacionais Abertos Usando Deep Learning

Murilo Gleyson Gazzola

<sup>1</sup>Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo (USP)  
Caixa Postal 668 – 13560-970 – Sao Carlos – SP

mgazzola@icmc.usp.br, gazzola@alumni.usp.br

**Abstract.** *Open Educational Resources (OER) are documents that are openly licensed and used for teaching, learning, and research purposes. They cover complete courses, textbooks, videos, softwares and any other tools, materials or techniques to support access to knowledge. The main difficulty, however, is to ensure the quality of these educational resources stored in online repositories. To fill this gap, a method was created using deep neural networks, specifically, a Recurrent Neural Network (RNN) to evaluation the quality of open educational resources, and compared with a Supporting Vector Machine (SVM) and its variations. The research methodology used was the use of an architecture for neural network, the creation of a controlled scenario, and the comparison with the main studies that perform automated evaluation of OER.*

**Resumo.** *Recursos Educacionais Abertos (REAs) são documentos abertamente licenciados e usados para fins de ensino, aprendizagem e pesquisa. Abrangem cursos completos, livros didáticos, vídeos, softwares e quaisquer outras ferramentas, materiais ou técnicas para apoiar o acesso ao conhecimento. A principal dificuldade, porém, é garantir a qualidade desses recursos educacionais armazenados em repositórios on-line. Para preencher esta lacuna, foi criado um método usando redes neurais profundas, especificamente, uma Rede Neural Recorrente (RNN) para avaliação automatizada da qualidade de recursos educacionais abertos, sendo comparado com uma Máquina de Vetores de Suporte (SVM) e suas variações. A metodologia de pesquisa utilizada foi a criação de uma arquitetura para rede neural, a criação de um cenário controlado, e a comparação com os principais trabalhos que realizam avaliação automatizada de REAs.*

## 1. Introdução

Recursos Educacionais Abertos (REAs) foram definidos por diversos trabalhos na literatura, sendo objeto de estudo de uma ampla diversidade de trabalhos [Wiley et al. 2014]. O termo foi cunhado em 2002 pela Unesco [UNESCO. 2002] que convocou o fórum sobre o impacto dos cursos abertos do ensino superior em países em desenvolvimento. Foi definido neste fórum que o termo “aberto” relacionava com recursos educacionais e indicava as possibilidades do conhecimento ser consultado, usado e adaptado por todos sem fins comerciais.

A ideia dos REAs é tornar o conhecimento do mundo um bem público tendo a tecnologia em geral, e especificamente a Web, como uma fonte de acesso a esse conhecimento. Entende-se, assim, que os REAs devem permitir seu compartilhamento, uso e

reuso. O movimento de REAs foi financiado inicialmente pela Fundação Hewlett que considerou o movimento uma filosofia de democratização do conhecimento por meio da Web [Smith and Casserly 2006] [Wiley et al. 2014].

Existem diversas definições de REAs, diferentes modelos de compartilhamento, diversos modelos de produção, como também muitos desafios para REAs. Neste trabalho, usamos a definição de REAs de [Gimenes et al. 2012] na qual *REAs podem ser caracterizados como materiais de ensino, aprendizagem e pesquisa, em qualquer meio de armazenamento, que estão amplamente disponíveis por meio de uma licença aberta que permite quatro liberdades mínimas conhecidas como 4R: Usar (Review), Aprimorar (Reuse), Recombinar (Remix) e Distribuir (Redistribute)*. Devido à importância do tema, em um documento recente [Miao et al. 2016] [Wiley et al. 2014], a Organização das Nações Unidas (ONU) definiu os principais problemas relativos ao desenvolvimento e ao uso de REAs: i) *o problema da qualidade* de REAs; ii) *o problema da descoberta*, isto é, como encontrar REAs, analisado anteriormente por [Gazzola et al. 2014]; iii) *o problema da sustentabilidade*, isto é, como financiá-los; iv) *o problema da localização e re-contextualização* de REAs; e v) *o problema do remix*, isto é, a dificuldade de identificar a granularidade de alteração do conteúdo por outras pessoas e o nível de alteração. De acordo com o documento da ONU, estes problemas devem ser enfrentados para que os REAs possam cumprir seu potencial de contribuir para o desenvolvimento humano.

A qualidade é de especial interesse para este artigo, pois pode ser abordada por meio de mecanismos computacionais [Bethard et al. 2009] [Custard and Sumner 2005] que avaliam tanto metadados como o conteúdo textual dos REAs. Este artigo dá os primeiros passos na utilização de novas abordagens de aprendizagem de máquina e apresenta uma arquitetura de *Deep Learning*, desenvolvida para avaliar a qualidade automática de metadados dos REAs. Foi realizada uma análise comparativa entre os métodos tradicionais de aprendizagem de máquina e o método proposto que usa uma arquitetura de *Deep Learning*.

*Deep Learning* ou Aprendizagem Profunda refere-se à presença de muitas camadas na rede neural artificial. Mas, esse significado mudou ao longo do tempo [Bengio et al. 2009, LeCun et al. 2015, Draelos et al. 2017]. Em 2012, 10 camadas já eram suficientes para considerar uma rede neural como profunda. Contudo, em 2017 é mais comum considerar uma rede neural profunda quando esta possui mais de 100 camadas [Draelos et al. 2017]. *Deep Learning* foi aplicado em diversos domínios como imagem, texto, vídeo, pronúncia e visão, melhorando significativamente os melhores resultados alcançados em dezenas de anos. Além disso, o sucesso de *Deep Learning* também se deve à maior disponibilidade de dados para treinamento e o custo baixo para utilização de GPUs (*Graphics Processing Unit*) para computação numérica eficiente. É possível citar a utilização de GPUs com custos baixos [Wan et al. 2014] as plataformas do Google, Microsoft e Amazon.

Nesta seção foram apresentados os conceitos básicos sobre REAs e a diferença entre aprendizagem de máquina tradicional e *Deep Learning*. As seções que seguem são organizadas da seguinte maneira. Na Seção 2 são apresentados os trabalhos correlatos desta proposta. Na Seção 3 são apresentados os conceitos fundamentais de redes neurais recorrentes e suas variações. Na Seção 4 é descrita a arquitetura desenvolvida nesta proposta. Em seguida, na Seção 5, é apresentada a metodologia de avaliação manual da

**Tabela 1. Tabela dos trabalhos que avaliam a qualidade de recursos educacionais automaticamente**

Trabalhos Relacionados	Material Composto do <i>Dataset</i>	Método de Avaliação Utilizado
Bethard et al. (2009)	REAs	SVM
Leary et al. (2011)	REAs	SVM
Cechinel et al. (2011)	OA	Perceptron Multilayer
Moise et al. (2011)	REAs	Perceptron e SVM
Wetzler et al. (2011)	REAs	SVM
Ahmed e Fuge (2017)	Desafios abertos	SVM e KNN

qualidade dos metadados de REAs. Por fim, na Seção 6 são feitas as análises sobre os resultados e na Seção 7 são feitas as conclusões e descritos os trabalhos futuros.

## 2. Trabalhos Relacionados

Os trabalhos relacionados a esta proposta possuem uma semelhança: quase todos utilizaram métodos tradicionais de aprendizagem de máquina. Além disso, o uso de SVM se mostrou superior às redes Perceptron. Contudo, nenhum trabalho utilizou técnicas de *Deep Learning* para avaliar a qualidade automática de REAs. A tabela comparativa (Tabela 1) mostra que trabalhos de avaliação da qualidade de REAs usam técnicas bastante comuns em Inteligência Artificial e focam mais esforços na arquitetura dos seus sistemas e na avaliação dos recursos utilizados. Não encontramos avaliação da qualidade de REAs via algoritmos de aprendizagem de máquina mais atuais, usando *Deep Learning*.

Os trabalhos de [Bethard et al. 2009] [Leary et al. 2011] [Moise et al. 2011] [Wetzler et al. 2013] apresentam avaliadores de qualidade de REAs usando SVM como classificador e arquiteturas com diferentes indicadores de qualidade. Enquanto, [Cechinel et al. 2011] apresenta um classificador de qualidade automática para Objeto de Aprendizagem (OA) e [Ahmed and Fuge 2017] cria uma forma de avaliar desafios on-line abertos, também, com uso do SVM como melhor resultado. Porém, nenhum deles tratam especificamente de redes neurais profundas para avaliação da qualidade.

## 3. Rede Neural Recorrente e a variante *Long Short Term Memory*

Uma Rede Neural Recorrente (RNN) é uma classe de redes neurais que exploram a natureza sequencial da sua entrada [Wu et al. 2017]. As RNNs têm sido aplicadas em várias tarefas como: reconhecimento de dígitos [Graves and Schmidhuber 2009], geração de texto [Sutskever et al. 2011], criação de modelos de língua [Mikolov 2012, Graves 2013, Pascanu et al. 2013], reconhecimento de voz [Graves et al. 2013], tradução automática [Bahdanau et al. 2014, Cho et al. 2014, Sutskever et al. 2014] e inferência textual [Bowman et al. 2015, Rocktäschel et al. 2015]. Uma RNN pode ser imaginada como um grafo de células (Figura 1), em que cada célula executa a mesma operação em cada elemento da sequência. Na Figura 1, cada elemento da sequência é representada por

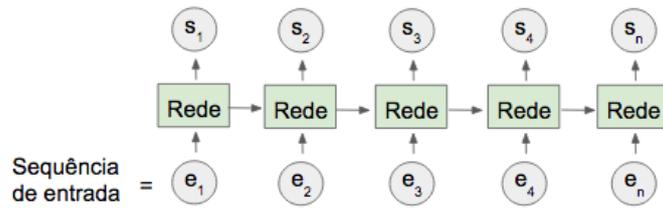


Figura 1. Arquitetura de uma RNN

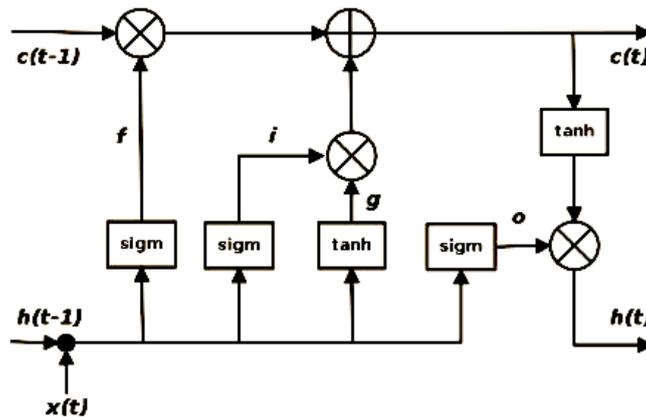


Figura 2. Arquitetura de uma LSTM [Donahue et al. 2015]

$e_n$ , em que  $n$  é o tamanho máximo da sequência de entrada. Além disso, o componente Rede representa parte de uma rede. Essa rede pode ser, por exemplo, uma RNN-Simples. O  $S_n$  representa cada ocorrência após ter passado por parte da Rede. As principais arquiteturas de RNNs são a arquitetura de rede Memória de Curto Prazo (LSTM, do inglês *Long Short-Term Memory*) e a arquitetura de Unidade Recorrente Fechada (GRU, do inglês *Gated Recurrent Units*). Essas duas arquiteturas não são as únicas, mas são as arquiteturas de RNNs mais usadas [Jozefowicz et al. 2015] para os principais problemas de sequência.

Uma LSTM é uma variante da RNN, sendo capaz de aprender dependências de longo prazo. A LSTM foi proposta pela primeira vez por [Hochreiter and Schmidhuber 1997] e refinada por outros pesquisadores ao longo do tempo [Donahue et al. 2015]. Uma LSTM é representada pela Figura 2. A linha na parte superior do diagrama é o estado da célula  $c$  que representa a memória interna da unidade. A linha em baixo é o estado oculto, e os *gates*  $i$ ,  $f$ ,  $g$  e  $o$  são o mecanismo pela qual a LSTM funciona em torno do problema de gradiente descendente [Andrychowicz et al. 2016]. Durante o treinamento, a LSTM aprende os parâmetros para esses *gates*. A fim de obter uma compreensão mais profunda de como esses *gates* modulam o estado oculto da LSTM, considere as equações que mostram como ela calcula o estado oculto  $h_t$  no tempo  $t$  é mostrado a seguir:

$$c_t = (c_{t-1} \otimes f) \oplus (g \otimes i)$$

$$h_t = \tanh(c_t) \otimes o$$

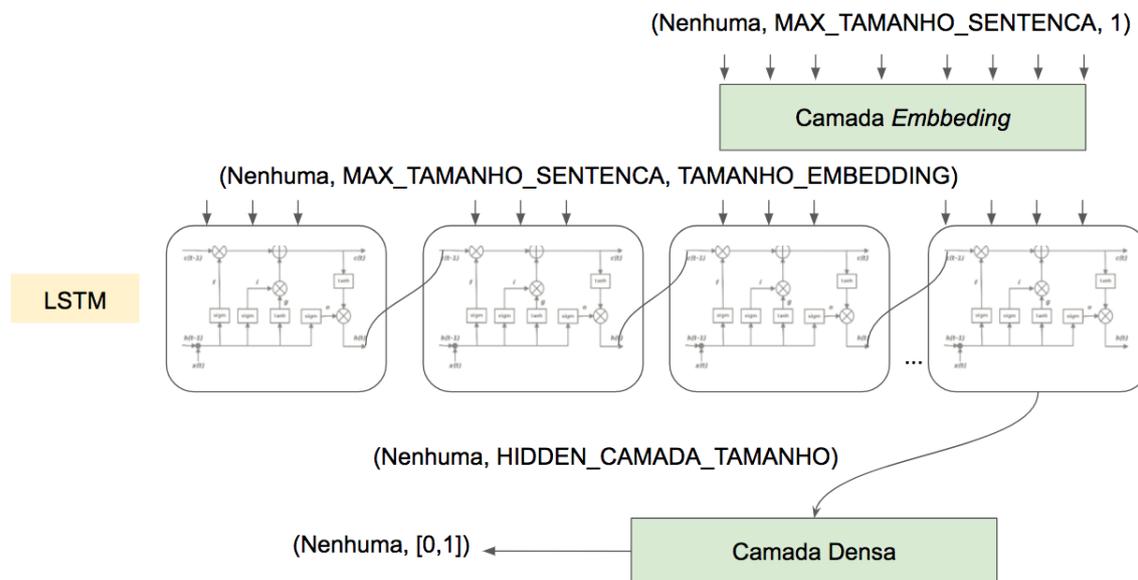


Figura 3. Arquitetura da proposta

#### 4. Arquitetura proposta

A Figura 3 representa a arquitetura desenvolvida. A entrada para cada linha é uma sequência de índices de palavras. O comprimento da sequência é dado por `MAX_TAMANHO_SENTENCA`. A primeira dimensão do tensor é definida como `Nenhuma`, para indicar o tamanho do lote. O tamanho do lote é o número de registros a cada vez que a rede é alimentada. Portanto, definimos a entrada da rede como  $(\text{Nenhuma}, \text{MAX\_TAMANHO\_SENTENCA}, 1)$ . Os pesos são iniciados com valores aleatórios pequenos e aprendidos durante o treinamento. Esta camada transformará o tensor em uma forma  $(\text{Nenhuma}, \text{MAX\_TAMANHO\_SENTENCA}, \text{TAMANHO\_EMBEDDING})$ . A saída da camada de incorporação é alimentada em uma LSTM com comprimento `MAX_TAMANHO_SENTENCA` e tamanho de camada de saída `HIDDEN_CAMADA_TAMANHO`. Então, a saída da LSTM é um tensor da forma  $(\text{Nenhuma}, \text{HIDDEN\_CAMADA\_TAMANHO}, \text{MAX\_TAMANHO\_SENTENCA})$ . Por padrão, a LSTM emitirá um único tensor da forma  $(\text{Nenhuma}, \text{HIDDEN\_CAMADA\_TAMANHO})$  em sua última sequência como  $(\text{Nenhuma}, [0, 1])$ . A saída da camada densa tem uma função de ativação Sigmoid, então ela emitirá 0 (REA sem qualidade) ou 1 (REA com qualidade).

Também, compilamos o modelo usando a função de perda e a entropia cruzada (*cross-entropy*) [De Boer et al. 2005] uma vez que prevê um valor binário. Além disso, utilizamos o otimizador Adam [Kingma and Ba 2014]. Por fim, treinamos a rede por 10 épocas e cada época validamos o modelo usando os dados do conjunto de teste.

#### 5. Avaliação Manual da Qualidade dos REAs

O *dataset* de REAs utilizado possui 400 REAs, totalizando 21.043 termos únicos e o máximo de elementos de cada vetor é de 4.495 termos. Portanto, utilizamos um máximo de *features* de 21.043 e o máximo da sentenças sendo 4.495. O *dataset* foi separado em dois conjuntos, o primeiro conjunto com 80% dados de treinamento e o segundo conjunto com 20% de dados de teste.

Os metadados foram avaliados manualmente por um anotador externo que considerou **Título, URL e Resumo** dos REAs. Esses metadados já foram extraídos e integrados no *dataset de metadados* por [Gazzola 2015] e disponível em repositório <sup>1</sup>.

Avaliação manual é composta de duas etapas: (1) ler todos os metadados e (2) verificar os seguintes critérios apresentados a seguir:

- O avaliador deve verificar se o título possui reconhecibilidade [Medelyan et al. 2009] [Matsune 2007]. A reconhecibilidade é uma descrição do assunto que trata o REA; mesmo uma pessoa que não é especialista da área do material consegue reconhecer. Utilizamos o conceito de reconhecibilidade, pois é fácil a classificação manual para diversos assuntos e não necessita de especialistas para cada área. Por exemplo: “Introdução à Ecologia e Biologia Evolutiva” é um título reconhecível, “ECE 420 Fall 2013” é um título não reconhecível.
- O avaliador deve identificar 5 elementos retóricos na estrutura do resumo. Os elementos a serem identificados são: a fundamentação teórica sobre o assunto do REA, propósito do REA, metodologia usada, resultados e conclusão. Caso observar pelo menos 3 elementos, o resumo deve ser considerado qualificado.
- O anotador deve verificar se a URL está presente nos metadados, indicando (SIM/NÃO).

Portanto, o REA será considerado não qualificado se algum critério de qualidade for desrespeitado. Caso contrário, o REA deve ser considerado qualificado. Por exemplo, se não possuir um título legível é considerado desqualificado ou se não possuir no mínimo 3 elementos retóricos do resumo dentre os 5 elementos citados anteriormente, ele é considerado não qualificado.

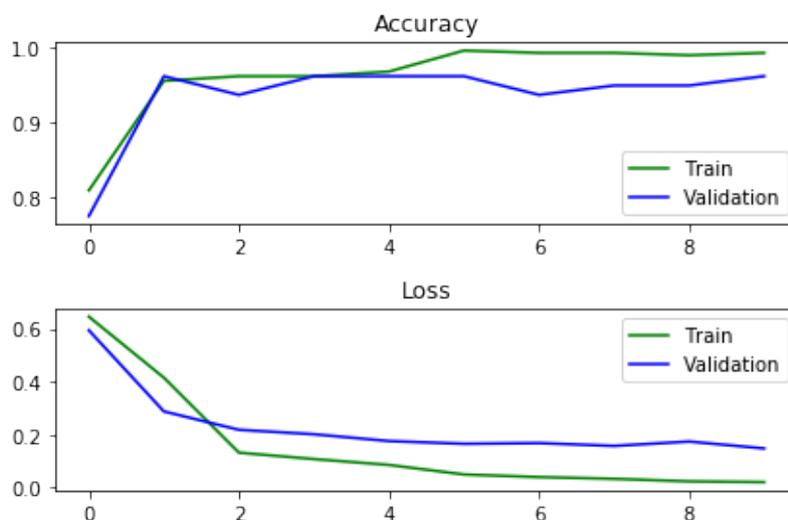
## 6. Análise dos Resultados

Os experimentos foram desenvolvidos em um ambiente controlado, foi usado como métrica para avaliação dos classificadores a acurácia (*Accuracy*) [Forman 2003] e a função perda (*Loss*). Para efeito de comparação entre os classificadores, foi usado o método SVM, pois diversos trabalhos de avaliação automática de REAs utilizam esse método como avaliador automático de qualidade [Bethard et al. 2009, Dalip et al. 2011, Cechinel et al. 2011, Ahmed and Fuge 2017].

Para entrada do método SVM, como também da nossa proposta, foi necessário construir um vetor de características. Para isso, escolhemos o modelo TF-IDF (*Term Frequency - Inverse Document Frequency*) [Ramos et al. 2003]. O modelo TF-IDF gera uma representação eficaz dos vetores textual na forma vetorial. A ideia é simples, quanto maior a frequência de um termo em um documento, mais representativo ele é para o conteúdo e quanto mais documentos contiverem um termo, menos discriminante ele é para o conteúdo.

A Tabela 2 mostra os resultados. Foram realizadas algumas otimizações conhecidas como *Truque do Kernel* [Jakkula 2006] que troca o Kernel para obtenção de resultados mais precisos. Para os experimentos foram testados empiricamente 3 variações: RBF, Linear e LinearSVC do *scikit-learn* [Scikit-learn 2017]. Na Figura 4 são apresentados os resultados da proposta desenvolvida: o primeiro gráfico o **eixo x** são as épocas que

<sup>1</sup>[https://github.com/gazzola/dataset\\_oer](https://github.com/gazzola/dataset_oer)



**Figura 4. Resultados da acurácia (*accuracy*) e da função perda (*loss*) da proposta desenvolvida**

**Tabela 2. Comparação entre a arquitetura proposta e os métodos tradicionais de aprendizado de máquina**

Algoritmos	Acurácia (%)
SVM-RBF	64.00
SVM-Linear	93.00
SVM-LinearSVC	95.00
<b>Proposta desenvolvida</b>	<b>96.25</b>

foram executadas e o **eixo y** são os valores da acurácia. No gráfico ( Figura 4) é mostrado uma comparação entre os dados de treinamento e os dados de validação descrito na Seção 5. No segundo gráfico (Figura 4) é mostrada a função de perda (*Loss*). O **eixo x** são as épocas e no **eixo y** são os valores de perda. Por fim, é possível observar que a proposta desenvolvida (Tabela 2) obteve um resultado superior ao SVM e suas otimizações.

## 7. Conclusões

Portanto, com esses experimentos pode-se concluir que o uso de *Deep Learning* é bastante vantajoso, suplantando métodos tradicionais da área de aprendizagem de máquina. Nesta arquitetura desenvolvida usando uma LSTM foi possível observar um avanço significativo apenas com uso do metadados. O intuito desses experimentos foi demonstrar o potencial do uso de *Deep Learning* no contexto de classificação e avaliação automática de REAs. Os próximos passos a serem desenvolvidos é a avaliação automática de todo o conteúdo dos REAs e não apenas dos seus metadados.

## Agradecimentos

A Capes pelo auxílio a bolsa de doutorado nº PROEX-8436630/D, ao CNPq e ao Núcleo Interinstitucional de Linguística Computacional (NILC-ICMC-USP).

## Referências

- [Ahmed and Fuge 2017] Ahmed, F. and Fuge, M. (2017). Capturing winning ideas in on-line design communities. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing, CSCW '17*, pages 1675–1687, New York, NY, USA. ACM.
- [Andrychowicz et al. 2016] Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., and de Freitas, N. (2016). Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, pages 3981–3989.
- [Bahdanau et al. 2014] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- [Bengio et al. 2009] Bengio, Y. et al. (2009). Learning deep architectures for ai. *Foundations and trends in Machine Learning*, 2(1):1–127.
- [Bethard et al. 2009] Bethard, S., Wetzer, P., Butcher, K., Martin, J. H., and Sumner, T. (2009). Automatically characterizing resource quality for educational digital libraries. In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pages 221–230. ACM.
- [Bowman et al. 2015] Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.
- [Cechinel et al. 2011] Cechinel, C., Sanchez-Alonso, S., and Garcia-Barriocanal, E. (2011). Statistical profiles of highly-rated learning objects. *Comput. Educ.*, 57(1):1255–1269.
- [Cho et al. 2014] Cho, K., van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. In *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111. Association for Computational Linguistics.
- [Custard and Sumner 2005] Custard, M. and Sumner, T. (2005). Using machine learning to support quality judgments. *D-Lib Magazine*, 11(10):1082–9873.
- [Dalip et al. 2011] Dalip, D. H., Gonçalves, M. A., Cristo, M., and Calado, P. (2011). Automatic assessment of document quality in web collaborative digital libraries. *Journal of Data and Information Quality (JDIQ)*, 2(3):14.
- [De Boer et al. 2005] De Boer, P.-T., Kroese, D. P., Mannor, S., and Rubinstein, R. Y. (2005). A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67.
- [Donahue et al. 2015] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Draelos et al. 2017] Draelos, T. J., Miner, N. E., Lamb, C. C., Cox, J. A., Vineyard, C. M., Carlson, K. D., Severa, W. M., James, C. D., and Aimone, J. B. (2017). Neurogenesis deep learning: Extending deep networks to accommodate new classes. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 526–533. IEEE.

- [Forman 2003] Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3(Mar):1289–1305.
- [Gazzola 2015] Gazzola, M. G. (2015). Uma arquitetura para mecanismos de buscas na web usando integração de esquemas e padrões de metadados heterogêneos de recursos educacionais abertos em repositórios dispersos. In *Dissertação de Mestrado - Instituto de Ciências Matemáticas e de Computação*. Acesso em: 2017-04-17.
- [Gazzola et al. 2014] Gazzola, M. G., Ciferri, C. D., and Gimenes, I. M. (2014). Seeoer: Uma arquitetura para mecanismo de busca naware por recursos educacionais abertos. In *Anais do Simpósio Brasileiro de Informática na Educação*, volume 25, page 1013.
- [Gimenes et al. 2012] Gimenes, I. M., Barroca, L., and Feltrim, V. D. (2012). Tendências na educação a distância e educação aberta na computação. *CSBC*.
- [Graves 2013] Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.
- [Graves et al. 2013] Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE.
- [Graves and Schmidhuber 2009] Graves, A. and Schmidhuber, J. (2009). Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 545–552.
- [Hochreiter and Schmidhuber 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- [Jakkula 2006] Jakkula, V. (2006). Tutorial on support vector machine (svm). *School of EECS, Washington State University*, 37.
- [Jozefowicz et al. 2015] Jozefowicz, R., Zaremba, W., and Sutskever, I. (2015). An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2342–2350.
- [Kingma and Ba 2014] Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [Leary et al. 2011] Leary, H., Recker, M., Walker, A., Wetzler, P., Sumner, T., and Martin, J. (2011). Automating open educational resources assessments: a machine learning generalization study. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, pages 283–286. ACM.
- [LeCun et al. 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.
- [Matsune 2007] Matsune, H. (2007). Using headings to improve visual readability. <https://goo.gl/EXw1r4>. Acessado em: 2017-09-04.
- [Medelyan et al. 2009] Medelyan, O., Milne, D., Legg, C., and Witten, I. H. (2009). Mining meaning from wikipedia. *International Journal of Human-Computer Studies*, 67(9):716–754.
- [Miao et al. 2016] Miao, F., Mishra, S., and McGreal, R. (2016). *Open educational resources: policy, costs, transformation*. UNESCO Publishing.

- [Mikolov 2012] Mikolov, T. (2012). *Statistical Language Models Based on Neural Networks*. PhD thesis, Brno University of Technology.
- [Moise et al. 2011] Moise, G., Vladiu, M., and Constantinescu, Z. (2011). Maseco—a multi-agent system for evaluation and classification of oers and ocw based on quality criteria.
- [Pascanu et al. 2013] Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 1310–1318.
- [Ramos et al. 2003] Ramos, J. et al. (2003). Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142.
- [Rocktäschel et al. 2015] Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kočiský, T., and Blunsom, P. (2015). Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.
- [Scikit-learn 2017] Scikit-learn (2017). Scikit-learn - machine learning in python. <http://scikit-learn.org/stable/modules/svm.html>. Acessado em: 2017-09-04.
- [Smith and Casserly 2006] Smith, M. S. and Casserly, C. M. (2006). The promise of open educational resources. *Change: The Magazine of Higher Learning*, 38(5):8–17.
- [Sutskever et al. 2011] Sutskever, I., Martens, J., and Hinton, G. E. (2011). Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024.
- [Sutskever et al. 2014] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.
- [UNESCO. 2002] UNESCO. (2002). Forum on the impact of open courseware for higher education in developing countries:: final report.
- [Wan et al. 2014] Wan, J., Wang, D., Hoi, S. C. H., Wu, P., Zhu, J., Zhang, Y., and Li, J. (2014). Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 157–166. ACM.
- [Wetzler et al. 2013] Wetzler, P., Bethard, S., Leary, H., Butcher, K., Bahreini, S. D., Zhao, J., Martin, J. H., and Sumner, T. (2013). Characterizing and predicting the multifaceted nature of quality in educational web resources. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 3(3):15.
- [Wiley et al. 2014] Wiley, D., Bliss, T., and McEwen, M. (2014). Open educational resources: A review of the literature. pages 781–789.
- [Wu et al. 2017] Wu, C.-Y., Ahmed, A., Beutel, A., Smola, A. J., and Jing, H. (2017). Recurrent recommender networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM '17*, pages 495–503, New York, NY, USA. ACM.